

Single- and Multi-Task Architectures for Tool Presence Detection Challenge at M2CAI 2016

Andru P. Twinanda¹, Didier Mutter², Jacques Marescaux², Michel De Mathelin¹, and Nicolas Padoy¹

¹ ICube, University of Strasbourg, CNRS, IHU Strasbourg, France

² IRCAD, IHU Strasbourg, University Hospital of Strasbourg, France

1 Introduction

The tool presence detection challenge at M2CAI 2016 consists of identifying the presence/absence of seven surgical tools in the images of cholecystectomy videos. Note that the tool *presence* detection task does not involve any localization of the surgical tools in the image. In Fig. 1, we show the seven tools that are included in the *m2cai2016-tool* dataset [4]³. The training dataset, released on May 23 2016, consists of 10 cholecystectomy videos annotated with the tool presence at 1 fps; while the testing dataset, released on September 9 2016, consists of 5 videos.

Here, we propose to use deep architectures to perform the tool presence detection task. This work is based on our previous work [4], where we presented several network architectures to perform multiple recognition tasks on laparoscopic videos. The tasks are surgical phase recognition and tool presence detection. Ultimately, we proposed an architecture which is designed to jointly perform both tasks. In this work, we are using both single-task and multi-task networks to address the challenge.

2 Methodology

In previous work [4], we proposed two convolutional neural network (CNN) architectures to perform tool presence detection: ToolNet and EndoNet, shown in Fig. 2. ToolNet is designed to solely perform the tool presence detection task, while EndoNet is designed to jointly perform the phase recognition and tool presence detection tasks. In [4], it has been shown that the multi-task network performs slightly better than the single-task counterpart. However, the multi-task network requires both phase and tool presence annotations, which are not available in the *m2cai16-tool* dataset. In Section 3.1, we will explain how we conduct our experiments to cope with this limitation.

The tool presence detection task is performed by using the output of the `fc_tool` layer which contains 7 nodes (equal to the number of tools). The output of this layer corresponds to the confidences of the presence of the seven tools in the image. By applying thresholds to these confidences, we can determine the presence of the surgical tools in the image.

³ The dataset is available at the official web page of M2CAI 2016: <http://camma.u-strasbg.fr/m2cai2016/>

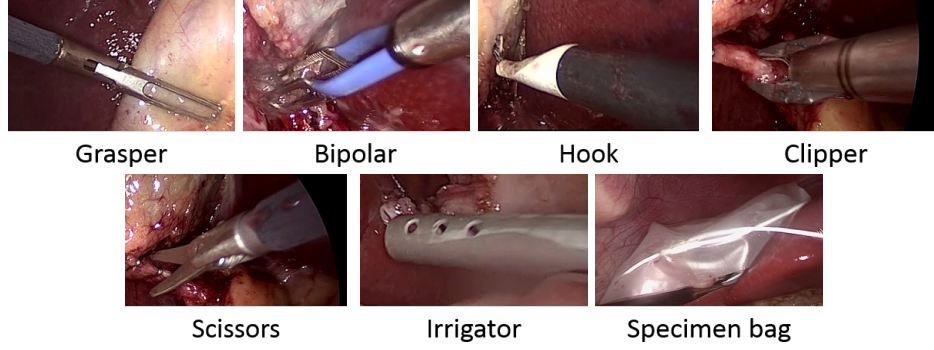


Fig. 1. Seven surgical tools included in the m2cai2016-tool dataset.

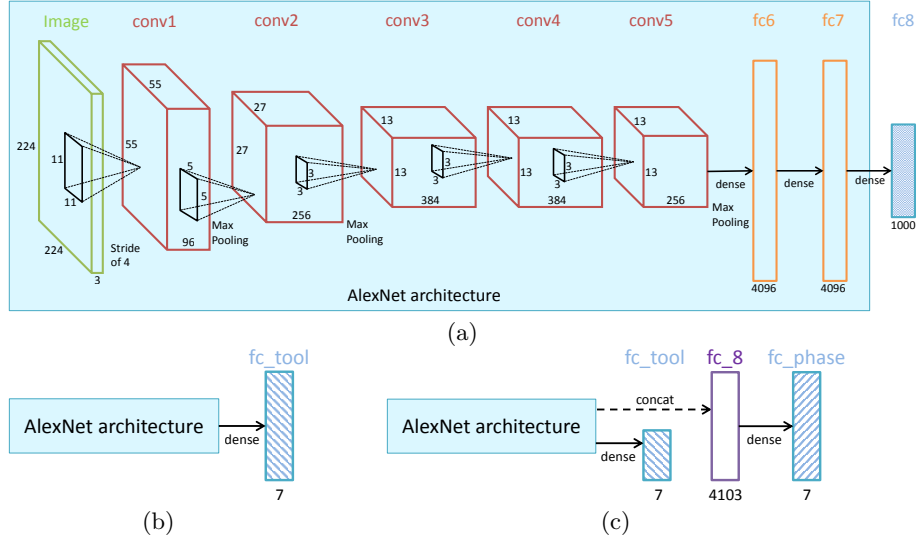


Fig. 2. The architectures of: (a) AlexNet, (b) ToolNet, and (c) EndoNet.

3 Experimental Setup

3.1 Network Comparison

As previously stated, the EndoNet architecture is designed to perform jointly surgical phase recognition and tool presence detection while the m2cai16-tool does not contain tool binary annotations. To cope with this limitation, we are using the Cholec80 dataset [4] which contains both phase and tool binary annotations. In addition to the extra annotation, the Cholec80 dataset contains more training videos than the m2cai16-tool dataset (i.e., 40 vs. 10 training videos).

Here, we will finetune multiple networks with the ToolNet and EndoNet architectures using m2cai16-tool and Cholec80.⁴

In summary, we are going to compare the performances of the following networks:

- ToolNet-m2cai16. This network is trained using the PhaseNet architecture on the m2cai16-workflow dataset;
- ToolNet-Cholec80. This network is trained using the PhaseNet architecture on the Cholec80 dataset;
- EndoNet-Cholec80. This network is trained using the EndoNet architecture on the EndoNet dataset.

3.2 ToolNet and EndoNet Finetuning Parameters

All networks are trained by fine-tuning the publicly available AlexNet network [2] which has been pre-trained on the ImageNet dataset [3]. The layers that are not defined in AlexNet (i.e., `fc_tool` and `fc_phase`) are initialized randomly. The network is fine-tuned for 50K iterations with $N_i = 50$ images in a batch. The learning rate is initialized at 10^{-3} for all layers, except for `fc_tool` and `fc_phase`, whose learning rate is set higher at 10^{-2} because of their random initialization. The learning rates for all layers decrease by a factor of 10 for every 20K iterations. The fine-tuning process is carried out using the Caffe framework [1].

3.3 Evaluation Metrics

The tool presence detection challenge is evaluated using mean average precision (mAP). This metric is obtained by computing the area under the precision-recall curve. The metric is first computed for each tool and then averaged over all the tools.

4 Experimental Results

We show the tool presence detection results in Table 1. It can be seen that ToolNet-m2cai16 does not perform very well, yielding an overall mAP of 52.5. This is significantly lower compared to ToolNet-Cholec80 and EndoNet-Cholec80 which yield 73.9 and 74.2 overall mAP, respectively. ToolNet-m2cai16 yields significantly lower mAP due to the fact the m2cai16-tool dataset contains significantly fewer training videos. This result has also been observed in the previous work [4] which showed that the system performance improves as the number of training size increases. Similarly to the results in [4], there is no significant improvement observed when the multi-task network is used for the tool presence detection task.

⁴ For more information regarding the Cholec80 dataset, we refer readers to [4].

Tool	ToolNet-m2cai16	ToolNet-Cholec80	EndoNet-Cholec80
Grasper	82.2	86.0	87.0
Bipolar	50.3	69.1	68.7
Hook	89.4	94.2	93.9
Scissors	17.0	51.9	52.8
Clipper	43.6	63.0	66.5
Irrigator	12.5	65.1	63.0
Specimen bag	72.2	88.6	87.3
MEAN	52.5±30.5	73.9±15.7	74.2±15.3

Table 1. Tool presence detection results.

Looking at the mAP of ToolNet-m2cai16 for each tool, one can notice the low recognition results for scissors, clipper, and irrigator. This is most likely due to the fact that these tools are only present during short period of times in the procedure. For this reason, the training images for these tools are scarce in the dataset. In addition, these tools have appearance similarities with other tools that appear very often during the procedures (i.e., grasper). Even when a larger training dataset is used to train the networks, there is still much room for improvements in the detection results of these three tools.

5 Conclusions

We have presented several approaches to address the tool presence detection challenge at M2CAI 2016. We proposed to use two types of CNN architectures to address the task: ToolNet and EndoNet. The former performs the tool presence detection task in a single-task manner, while the latter performs the task jointly with the phase recognition task. The results show that for tool presence detection, the multi-task architecture does not necessarily introduce high improvements to the results compared to the single-task counterpart. Instead, a significant improvement is obtained when there are more data available to train the networks. This improvement can be regarded as a call for action for other institutions to start working toward publishing more datasets into the community, so that better models could be generated to perform the task.

Here, the tool presence detection task is addressed in a frame-wise manner, i.e., there is no temporal information incorporated in the detection process. For future work, it would be interesting to see whether the temporal information plays a role in the detection process. To establish an end-to-end architecture, the temporal information can be incorporated with the usage of recurrent neural network (RNN).

References

1. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture

- for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
2. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012.
 3. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.
 4. Andru Putra Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel de Mathelin, and Nicolas Padoy. Endonet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Transactions on Medical Imaging*, 2016.